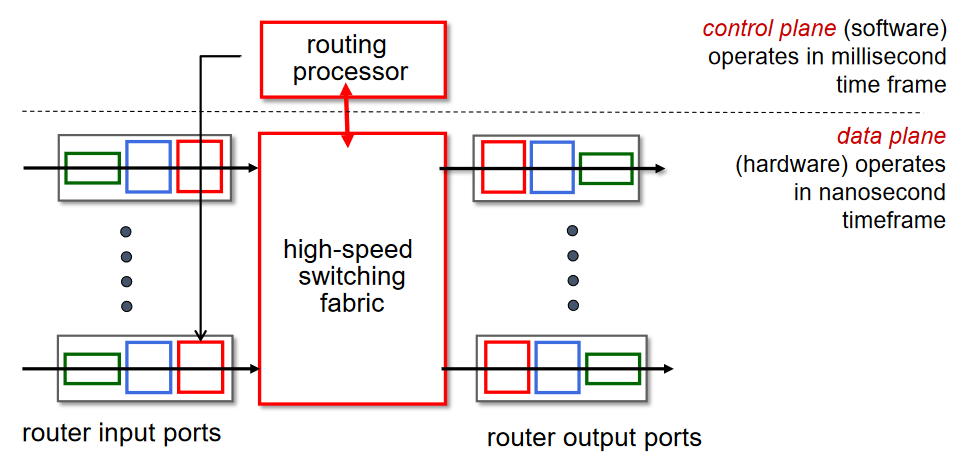
# Livello di Rete - Piano Dati

Abbiamo visto quindi 2 tipi di funzioni che il livello di rete implementa: forwarding (il pkg viene instradato da router a router) e il routing (determina il percorso generale che dovrà fare il pkg).

Il **Data Plane** è LOCALE ed è HardWare, è implementato nei singoli nodi/router. Stabilisce come un datagramma deve essere inoltrato, ovvero implementa il **forwarding**.

Il Livello di Rete è un BEST EFFORT SERVICE MODEL (di default in tutte le reti), ovvero non implementa il recupero dei pkg, la gestione degli errori o la garanzia della consegna.

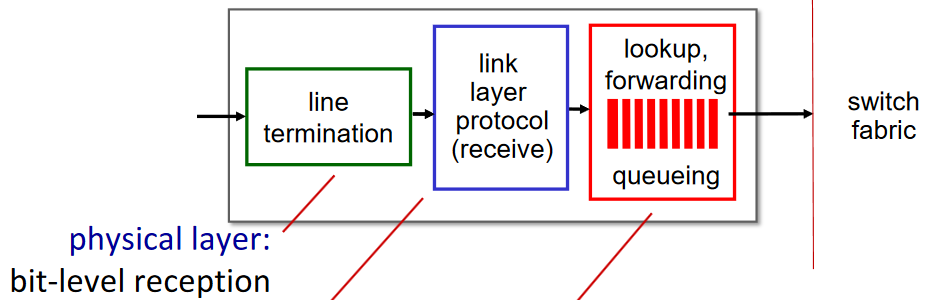
**ARCHITETTURA DEL ROUTER**



Ci sono tot. porte in input e in output suddivise LOGICAMENTE (ovvero le porte gestiscono I/O ma viene usata solo una funzionalità, o Input oppure Output).

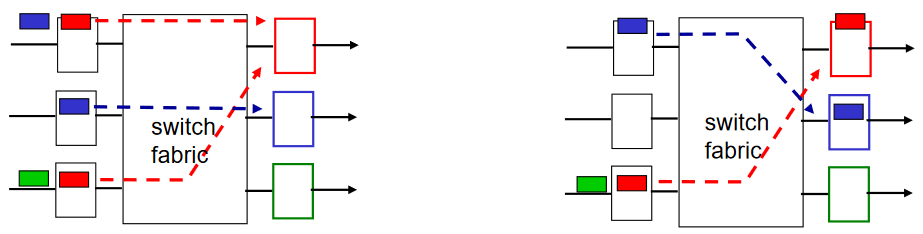
Switching Fabrics: trasferisce i pkg da input ad output, bisogna determinare la velocità di switching (idealmente uguale a N\*R, N = numero di input, R = tempo di smistamento, così non c'è accodamento).

**INPUT PORT FUNCTIONS**



Fa mettere in coda i datagrammi in attesa di essere smistati.

**Input Port Queueing**: Se la Switch Fabrics non riesce a smistare tutti i datagrammi che riceve allora si crea un problema di coda. **HOL Blocking** = il primo che arriva ha la priorità.



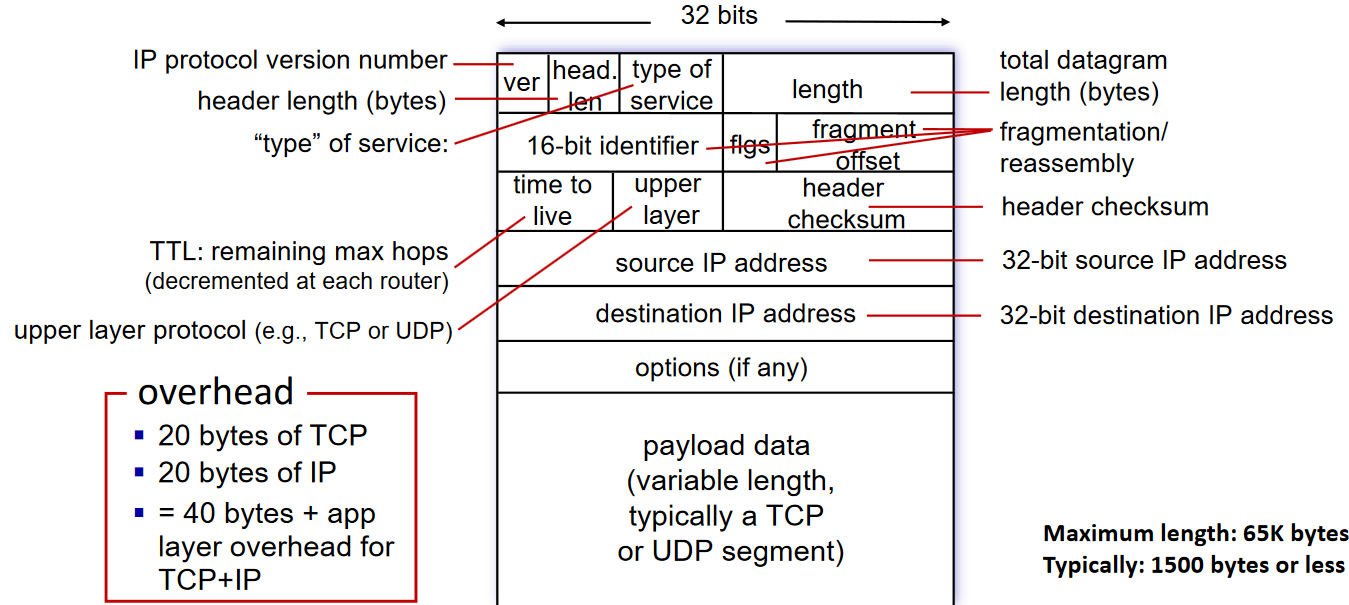
*Il verde nonostante abbia l’output libero deve aspettare che il rosso si si tolga di mezzo. Sta subendo HOL Blocking.*

**Output Port Queueing**: La Switch Fabrics è più veloce di R. Il Buffer si riempie e questo comporta l’applicazione di alcune politiche e quindi la perdita di pacchetti.

Drop Policies -> scarto datagrammi di bassa priorità. Scheduling Policies -> tra tutti i datagrammi chi devo trasmettere per primo?

Il Data Plane è quindi implementato a livello HW perchè deve operare ad altissime velocità (nanosecondi). L’HW non permette fisicamente di avere traffico, se 2 porte hanno problemi esse non influenzano le altre porte.

**IP DATAGRAM FORMAT**

****

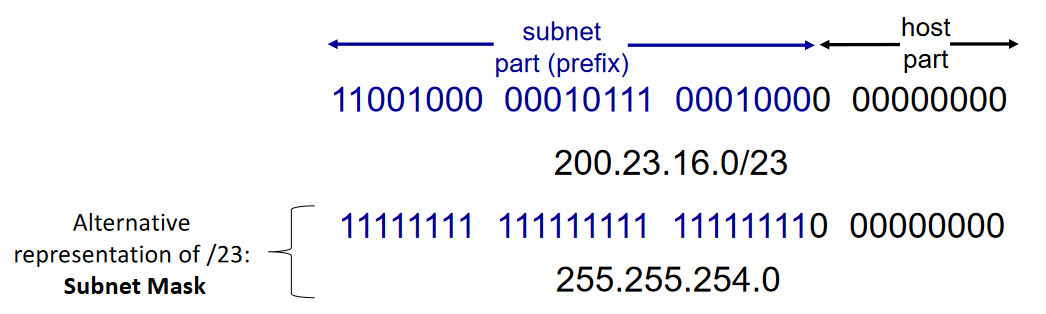
Protocollo IP = unico protocollo usato a livello di rete (32 bit), ogni host/interfaccia è associata ad un IP Address. La frammentazione non serve a un cazzo

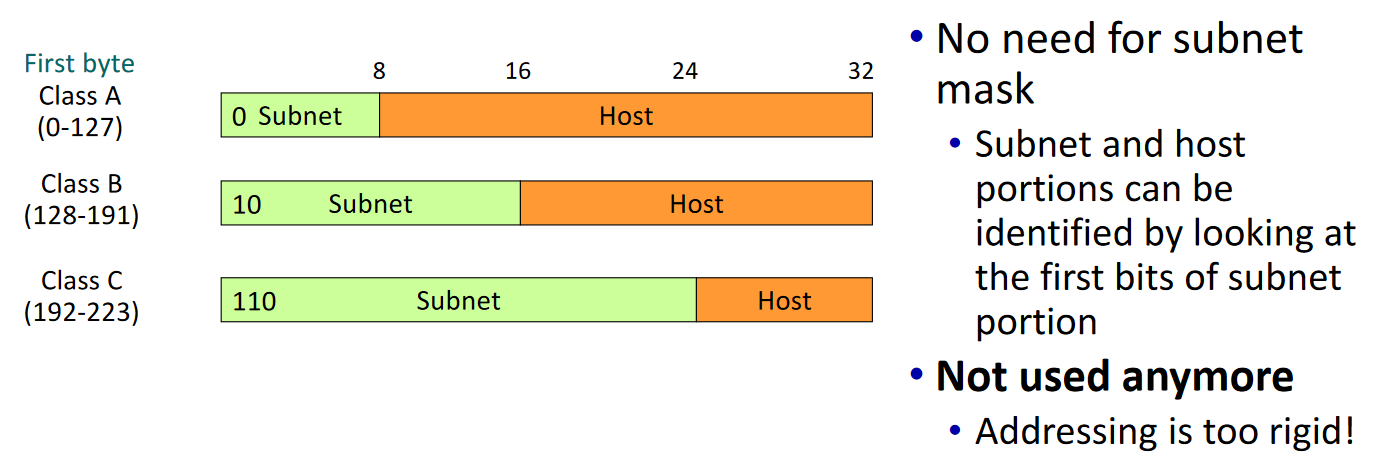
**IP ADDRESSING**

L’interfaccia è una connessione fisica tra 2 host/router. Ci sono interfacce collegate senza passare da un router, esse si chiamano sotto-reti.

IP diviso in Subnet-Part e Host-Part (parte più a destra). Le subnet sono sotto-reti, quindi host che non devono passare dal router per comunicare tra di loro.

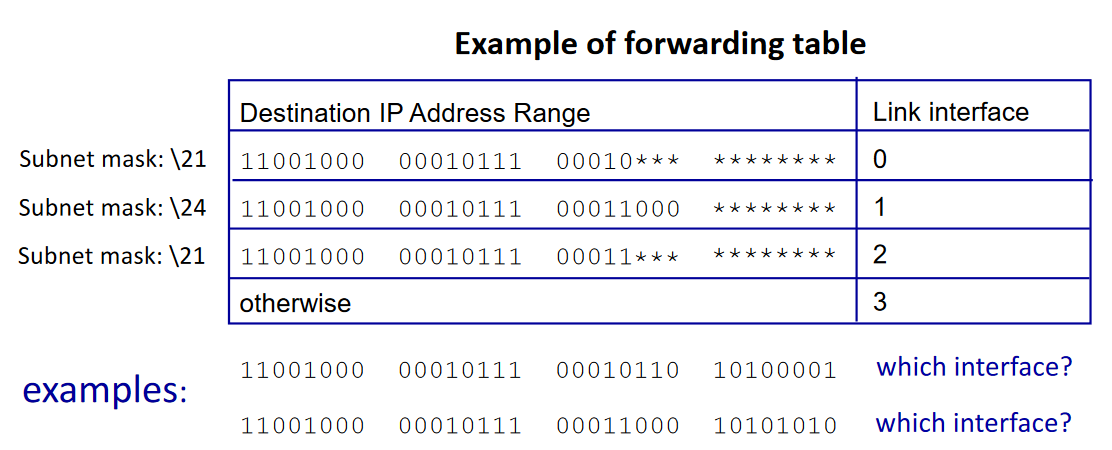
**CIDR** -> metodologia per separare subnet da host. /24 vuol dire 24 bit dedicati alla subnet e 8 bit dedicati all’host. La Subnet Mask è la riscrittura dello spazio dedicato alle subnet.

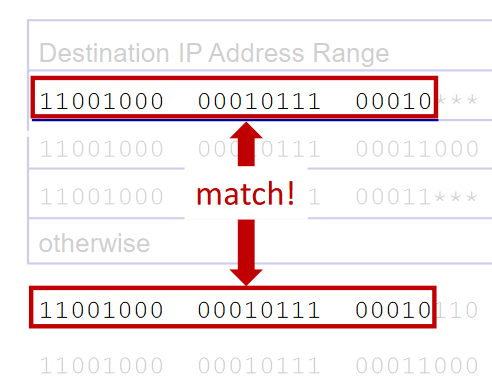


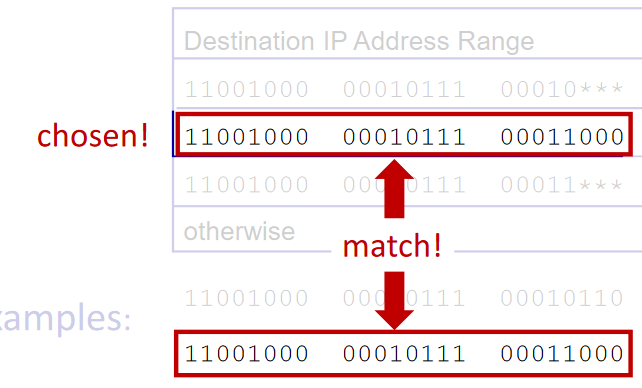


La classe A ha tanti host, poche sottoreti, viceversa per la classe C, che ha pochi host ma molte sottoreti

**DESTINATION-BASED FORWARDING**







Quali di questi 2 devo usare? Semplice, devo usare il prefisso più lungo, quindi la seconda immagine è quella giusta.

**DHCP**

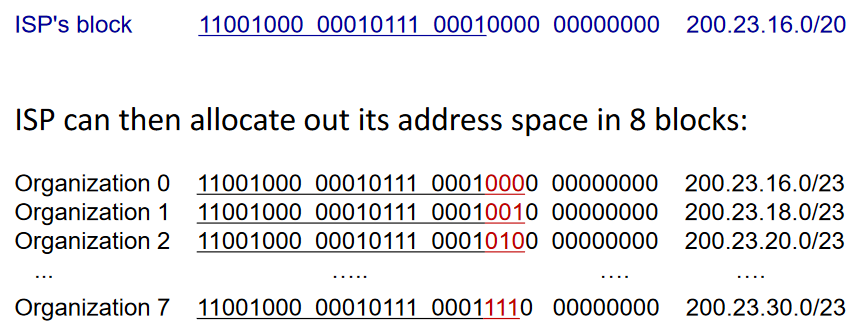
Un host può ricevere il suo indirizzo IP in modo statico (assegnato fisicamente) oppure attraverso il DHCP.

DHCP assegna dinamicamente un IP, permette di usare quelli non in uso

1. **DHCP DISCOVER**: L'host invia un messaggio di richiesta DHCP sulla rete. Il server DHCP è un altro dispositivo nella rete, che potrebbe essere il router domestico.
2. **DHCP OFFER**: Il server DHCP risponde con un messaggio di offerta DHCP, in cui propone un indirizzo IP disponibile per l'host (se vuole anche subnet mask e il server DNS). I primi 2 messaggi non servono se l’host si ricorda di aver già parlato con lo stesso DHCP Server.
3. **DHCP REQUEST**: L'host risponde con una richiesta DHCP, accettando l’indirizzo IP offerto.
4. **DHCP ACK**: Il server DHCP invia un messaggio di conferma DHCP all'host, confermando ufficialmente l'assegnazione dell'indirizzo IP.

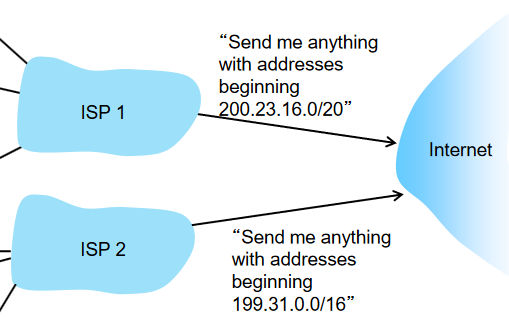
**INDIRIZZAMENTO GERARCHICO**

Come fa un indirizzo IP a essere suddiviso in parti, subnet e host? Ottiene una parte assegnata dello spazio degli indirizzi dal suo ISP provider. ISP ottiene gli IP da ICANN (azienda).



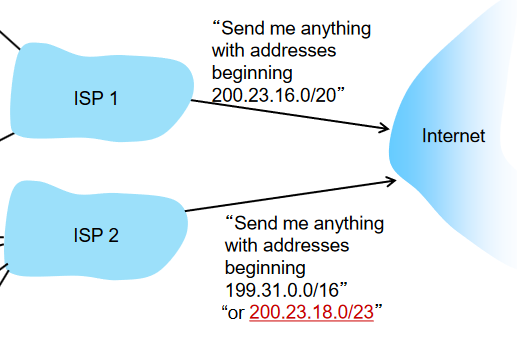
ISP ha una subnet di /20. Ma dato che vuole aumentare i guadagni suddivide ulteriormente l’IP dando più spazio alla subnet e meno agli host, facendo diventare la subnet = /23.

Questo può essere utile per fare aggregazione delle rotte, cioè con una sola riga nella tabella di inoltro (200.23.16.0/20) internet può inviare msg all’ISP e poi sono cazzi dell’ISP mandare il msg all’host giusto. Questo semplifica il lavoro del router.



In più se un host (200.23.18.0/23) cambia ISP basta aggiungere l’indirizzo IP alla tabella di inoltro e il gioco è fatto.

La tabella, se ricordi, sceglie il prefisso più lungo, per questo non si confonde con l’altro ISP.



**NAT**

Gli indirizzi IPv4 sono finiti ma vengono ancora utilizzati, questo grazie alla NAT. Essa permette a più dispositivi di condividere lo stesso indirizzo IP.

Come riconoscere i vari host se hanno tutti lo stesso IP? Attraverso le porte. Ma che vantaggi ci sono:

* just one “public” IP address needed from provider ISP for all devices
* can change addresses of host in local network without notifying outside world
* can change ISP without changing addresses of devices in local network
* security: devices inside local net not directly addressable, visible by outside world

Infatti gli IP pubblici usano la NAT, mentre gli indirizzi IP privati si usano solo nelle LAN

**NAT TRANSLATION TABLE**

Trasforma la coppia (IP Host, Porta) in (IP Lan, Porta) e viceversa.

Il NAT è odiato perchè il router dovrebbe processare solo fino al livello 3 del , il fatto che operi con le porte rompe questa premessa.

**INDIRIZZI IP SPECIALI**, non possono essere assegnati ad un host.

* Tutta la parte di host è 0 -> ~.~.~.0/24
* Indirizzo di broadcast diretto (mai usato) -> ~.~.~.255/24
* Indirizzo di broadcast limitato (usato spesso), tutta la parte di host è 1 -> .255.255.255.255/24